Brief del proyecto

Problema y usuarios principales

Encontrar ofertas laborales acordes al perfil de cada persona puede ser un proceso largo y tedioso. Los usuarios tienen que revisar una gran cantidad de publicaciones que no siempre se ajustan a sus intereses, nivel de experiencia o formación académica. Esto genera pérdida de tiempo, frustración y dificultad para mantenerse al tanto de nuevas oportunidades laborales.

Por lo tanto, nuestro chatbot está dirigido a aquellas personas que buscan activamente empleo, utilizan Telegram como canal de comunicación y valoran la automatización.

Objetivo del MVP

Las búsquedas iniciadas en el chatbot deben resultar en una lista formateada de ofertas laborales relevantes, seguida de un análisis y priorización por IA, o en un mensaje de error claro, sin caídas del servicio.

Suposiciones y riesgos

Suposiciones

- La ScrapingDog Jobs Search API es la fuente de datos principal y se mantiene accesible y operativa.
- La integración de la librería *python-telegram-bot* permite un flujo conversacional estable.

Riesgos

- Dependencia de API: La API de ScrapingDog podría cambiar su esquema o dejar de funcionar.
- Límites de uso de la API: La API de ScrapingDog (en modo gratuito o de prueba) impone límites de solicitudes. Un uso intensivo durante las pruebas o por múltiples usuarios podría bloquear el servicio.

PRD

Alcance

El MVP se centra en la interacción conversacional vía Telegram para la recolección de parámetros, la ejecución modular de la búsqueda de empleos, la presentación de los resultados y un análisis de IA de las ofertas.

Historias/escenarios de uso

- **Búsqueda completa:** Como usuario, quiero iniciar el bot (/start), ingresar mis 5 criterios de búsqueda paso a paso interactuando con botones para Tipo, Nivel y Modalidad, y recibir una lista formateada de ofertas laborales seguida de un análisis por la Inteligencia Artificial.
- **Cancelación del flujo:** Como usuario, quiero poder cancelar la secuencia de ingreso de parámetros en cualquier momento (ej. usando /cancel) para reiniciar la conversación más tarde.
- **No se encontraron resultados:** Como usuario, si mi búsqueda no arroja ofertas, quiero recibir un mensaje claro de "No se encontraron resultados" en lugar de un error.

Casos borde

- **Fallo de conexión (API):** La API de ScrapingDog no responde (error 500) o hay un fallo de red. El bot notifica el error (❌ Error al conectar con la API), y registra el fallo en log_errores.txt antes de finalizar la búsqueda.
- **Fallo en el Análisis IA (Gemini):** La API de Gemini no responde. El bot debe saltar este paso, notificar el error (⚠️ No se pudo contactar al módulo de Inteligencia Artificial), y finalizar la búsqueda, cumpliendo con la entrega de resultados crudos.

Funcionalidades v1

Categoría	Funcionalidad	Descripción
Must Have	Integración con Telegram	El bot debe estar accesible, en línea y en modo *polling* (operación de consulta constante), manejando el *token* para responder a mensajes.
Must Have	Flujo conversacional	Implementación del flujo para guiar al usuario secuencialmente por los 5 parámetros requeridos (eliminando el uso de input() de la versión de consola).

Must Have	Módulo Extractor (API)	La clase `ExtractorEmpleos` debe ejecutar la llamada a la API con los 5 parámetros recibidos.
Must Have	Presentación formateada	La clase Presentador debe formatear la lista de ofertas (Título, Empresa, Ubicación, Link) en formato HTML de Telegram para su correcta visualización.
Should Have	Botones interactivos	Usar botones *inline* de Telegram para seleccionar las opciones fijas (Tipo, Nivel, Modalidad), mejorando la UX y garantizando datos correctos.
Should Have	Manejo de excepciones	Implementar `try...except` para manejar fallos de red (`requests.exceptions.RequestException` y `HTTPError`) y registrar el error en un `log_errores.txt`.
Nice to Have	Análisis por IA (Gemini)	Integración con Gemini AI para analizar, resumir y priorizar las 3 a 5 mejores ofertas encontradas. La funcionalidad incluye la lógica de fragmentación de mensajes para entregar informes largos.

Criterios de aceptación

- Ready: Se han definido los prompts de cada estado de la conversación (las 5 preguntas) y se han creado los diccionarios de mapeo para convertir las respuestas del usuario a los valores de la API. Las clases Extractor, Procesador y Presentador están completas y modularizadas para ser llamadas desde el módulo de Telegram.

- Done:
    1. El bot de Telegram puede iniciarse y mantiene la conexión por polling sin caerse durante la prueba.
    2. El flujo de conversación se completa correctamente, se llama al Extractor y se muestran los resultados en Telegram.
    3. Se ejecuta correctamente la lógica de análisis por IA.
    4. En caso de no encontrar resultados, se maneja y se comunica correctamente.

Bosquejos rápidos

Elemento	Texto del Bot	Acción/Respuesta del usuario
Paso 1: Inicio	🤖 Bienvenido al Buscador de Empleos 💼\nComencemos con los parámetros de búsqueda.\n\n 🔹 Área o rubro de trabajo (ej: Python Developer):	/start (Comando inicial)
Paso 2:	Input Rubro	Data Analyst (Texto libre)
Paso 3:	Tipo de trabajo	🔹 Ahora, selecciona el Tipo de trabajo:
Paso 4:	Nivel de experiencia	🔹 ¿Cuál es tu Nivel de experiencia?
Paso 5:	Modalidad	🔹 ¿Cuál es la Modalidad de trabajo?
Paso 6:	Ubicación	🔹 Por último, la Ubicación (ej: Buenos Aires, Argentina):
Paso 7: Búsqueda	🔎 Buscando ofertas... Esto puede tardar unos segundos.	(El bot ejecuta la API y el procesamiento)
Paso 8: Resumen	💼 Primeras 5 de X ofertas encontradas:	(Ninguna)

Paso 9:	Análisis IA	🧠 Ahora, el análisis de las ofertas por Gemini AI...
Paso 10: Informe IA	🤖 Análisis de Empleos por Gemini: \n\n (Resumen y priorización de 3 a 5 ofertas por la IA. Si es necesario, se envía en múltiples mensajes.)	(Ninguna)
Paso 11: Fin	✅ Búsqueda finalizada. Usa /start para otra búsqueda.	(Ninguna)
No Resultados	❌ No se encontraron ofertas con esos criterios. Por favor, intenta con una búsqueda más amplia.	(El bot termina la conversación)
Error	❌ Error al conectar con la API de empleos. La cuota puede haberse agotado o hubo un fallo en el servicio.	(El bot termina la conversación)